

Language Server Protocol

Gyén Attila

Kódmegértést támogató eszközök

- Szoftver élekciklus
- Karbantarthatóság
- Fejleszthetőség
- Kódminőség
- Dokumentáció
- Újonnan bevont fejlesztők

IDE és a Kódszerkesztő

- Új kódok írása
- Kód kiegészítés
- Szintaxis kiemelés
- Szemantikai és szintaktikai hibaellenőrzés
- Egyes esetekben a kódon belüli navigáció

IDE vs. Kódszerkesztő

- IDE
 - Fordító
 - Hibakereső
 - Tesztelő
 - Lassabban áll fel a rendszer, minden indításkor idő megy el a kód indexelésére
- Kódszerkesztő
 - Nem minden esetben tartalmazza az IDE-nél említett dolgokat, cserébe
 - Gyorsabban elindul és kevesebb memóriát fogyaszt, kisebb projekteknél vagy, ha csak szerkeszteni kell a kódban előnyösebb lehet, mint az IDE

Miért jó ezeknek a használata?

- Megkönnyítik a fejlesztők mindennapi életét
- Felgyorsítják a kód írását és tesztelését
- Segítik az újonnan érkező fejlesztők minél gyorsabbi becsatlakozását a projektbe



AZONBAN!!

LSP

- Goto definition
- Cross-join implementáció kellene a különböző API-k miatt
- Egységesített API-t kínál
- Microsoft
- TypeScript

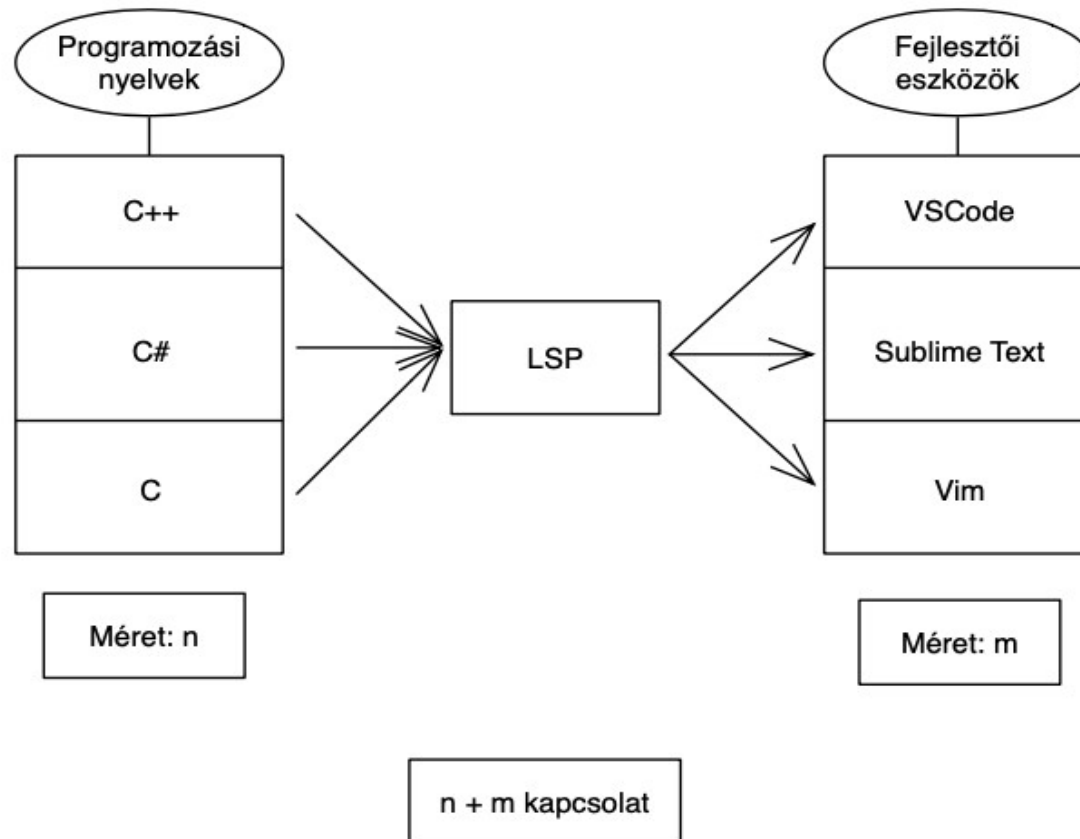
Nyelvi kiszolgáló szerverek

- Nagy mennyiségű adat kezelése egyidőben
- AST - Absztrakt szintaxisfa felépítése
- Statikus kódelemzés
- Magas a CPU és a memória használat

Miért éppen az LSP?

- Komplexitás csökkentés

Megoldás:

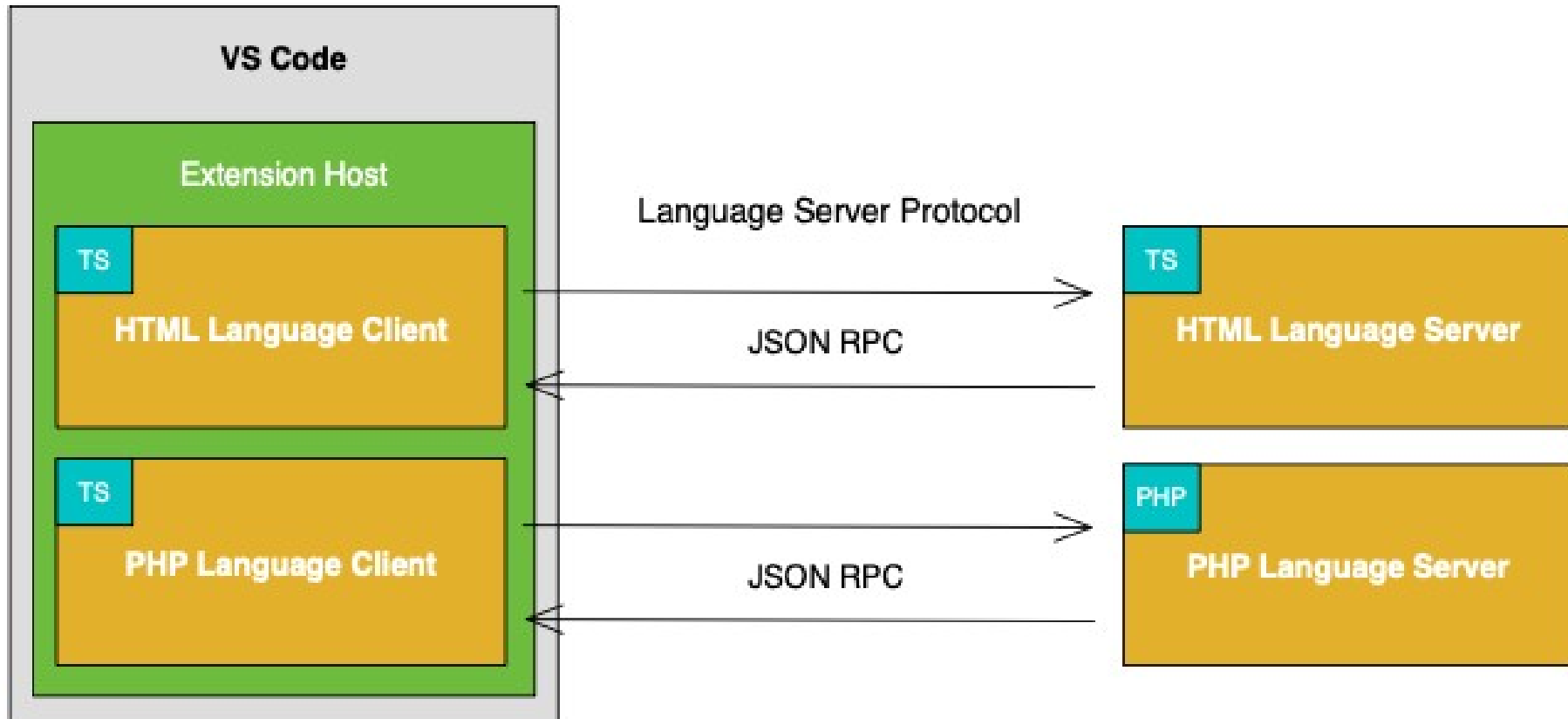


Miért éppen az LSP?

- Komplexitás csökkentés
- Teljes mértékben nyílt forráskód
- Független a nyelvi kiszolgálóktól ill. a fejlesztőieszközöktől

Implementáció

- Két részre osztható
 - LSP kliens
 - Nyelvi szerver
 - Elemző eszköz
 - Nyelvi elemző - külön folyamat

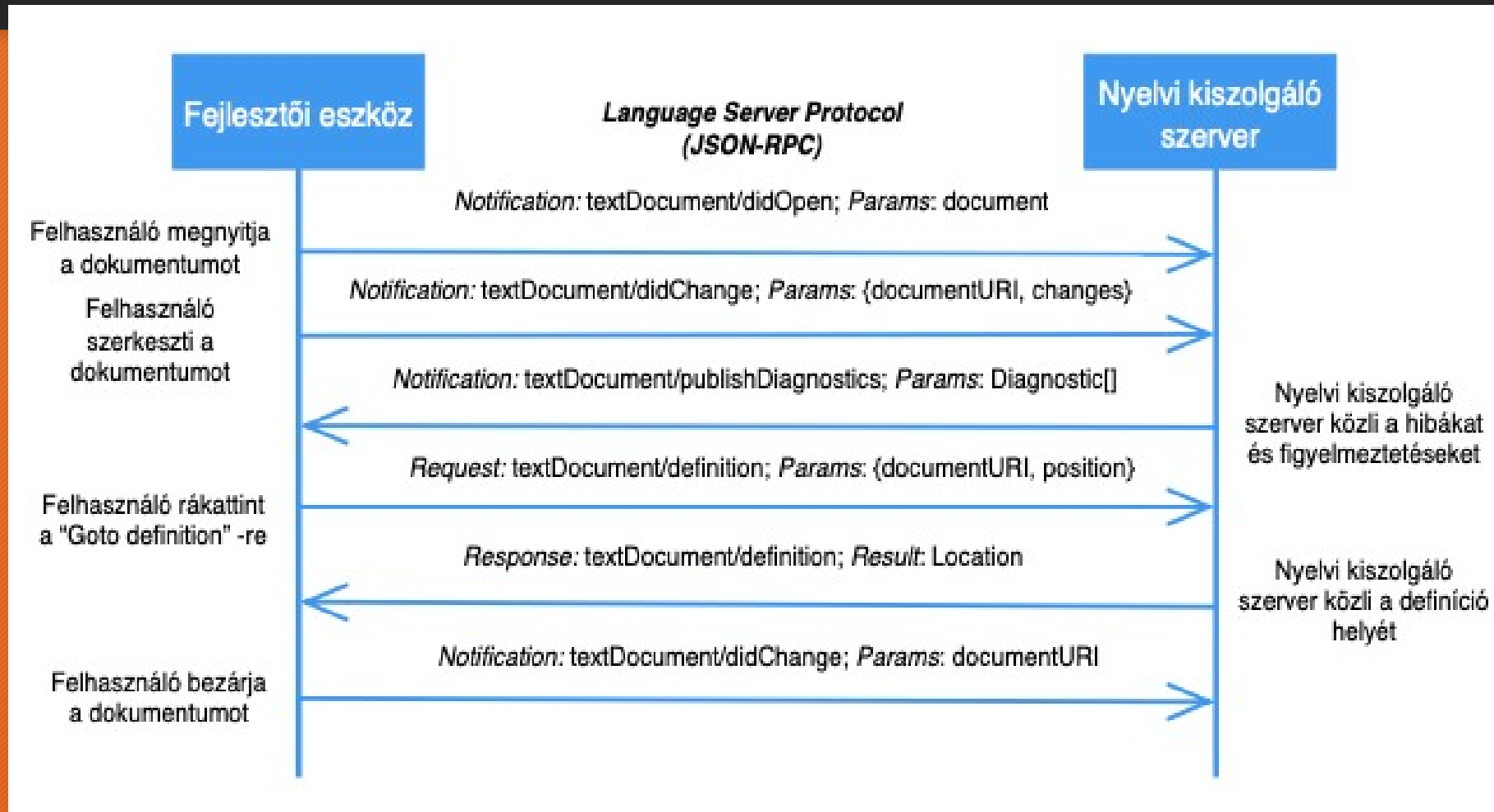


Specifikáció

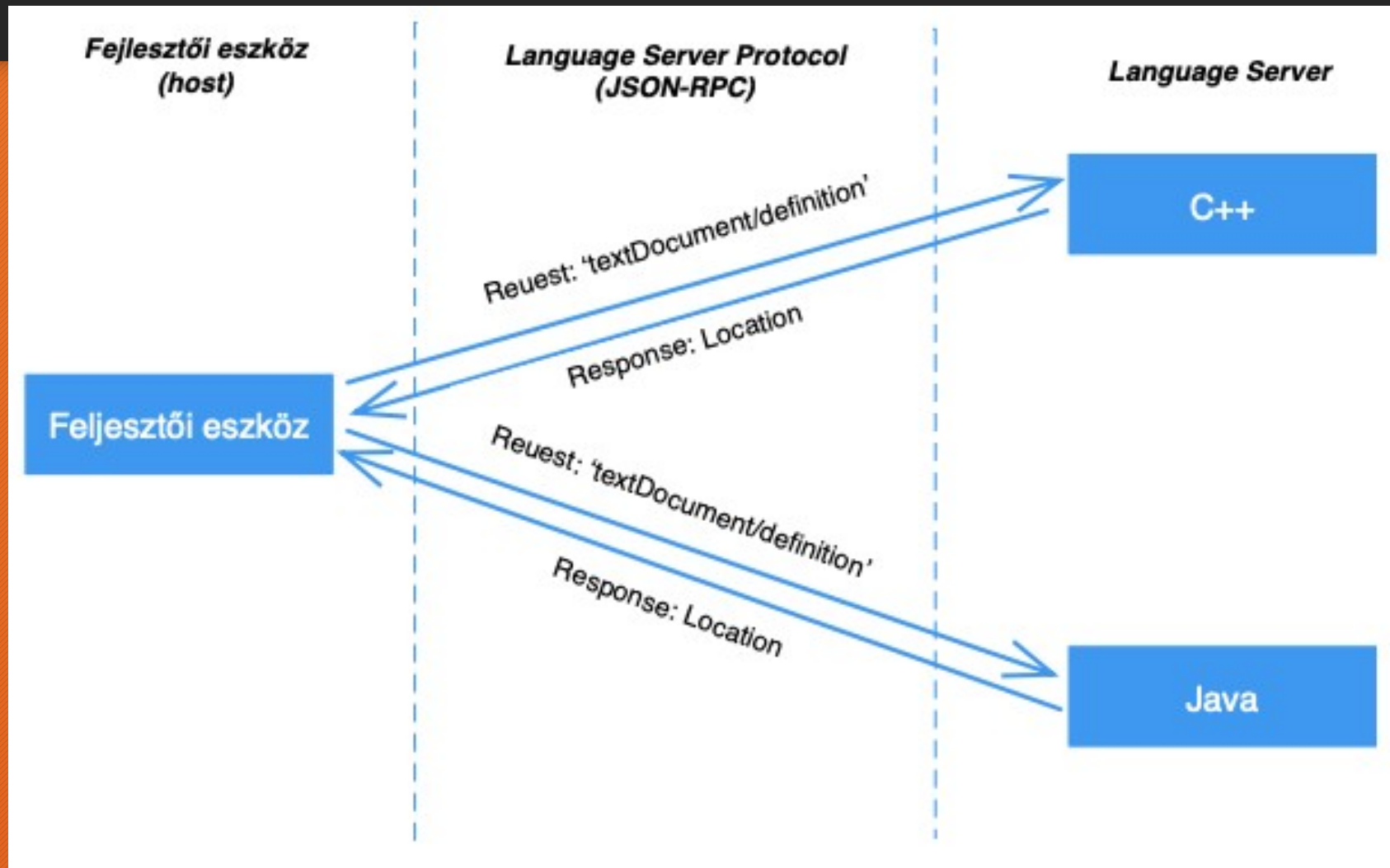
- Az alap protokoll
 - Fejléc rész - név + érték
 - Content-length
 - Content-type
 - Tartalmi rész
 - Klines \equiv \square Szerver
 - JSON-RPC

```
Content-Length: ... \r\n
\r\n
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "textDocument/didOpen",
  "params": {
    ...
  }
}
```

Működés



Külön nyelvi szerver indítás



LSP Kérések

- “GOTO” kérések
 - Declaration
 - Definition
 - Implementation
- Információ nyújtó kérések
 - Completion
 - Hover
 - Signature help

LSP Kérések

- “Document” kérések
 - Highlight
 - Symbol
 - Link
- Dokumentum módosító kérések
 - Formatting
 - Rename
 - Type formatting

Goto Implementation példa

```
connection.onInitilaize((params: InitializeParams) => {
    let capabilities = params.capabilities;
    // ...
    return {
        capabilities: {
            textDocumentSync: documents.syncKind,
            implementationProvider: true
        }
    };
});
```

Goto Implementation példa

```
connection.onImplementation(  
  (document: TextDocumentPositionParams): Definition => {  
    return Location.create(document.textDocument.uri, {  
      start: { line: 0, character: 0 },  
      end: { line: 0, character: 1 }  
    });  
  } );
```

Összegzés - LSP Pro Kontra

- Pro
 - Sokan fejlesztik
 - Implementáció komplexitásának csökkentése
 - Szabad felhasználás
 - Szabadon bővíthető
- Kontra
 - Nehéz dokumentációt találni a használatához
 - Elavult, hiányos
 - Sok pontnak kell eleget tenni ahhoz, hogy a saját implementációnk bekerüljön a hivatalos LSP repóba

Köszönöm...

... a figyelmet!
Várom a kérdéseket...